Concept-wise Fine-tuning Matters in Preventing Negative Transfer

Yunqiao Yang[1*], Long-Kai Huang[2], Ying Wei[1†]
[1]City University of Hong Kong [2]Tencent AI Lab
{hustyyq, hlongkai, judyweiying}@gmail.com

Abstract

A multitude of prevalent pre-trained models mark a major milestone in the development of artificial intelligence, while fine-tuning has been a common practice that enables pre-trained models to figure prominently in a wide array of target datasets. Our empirical results reveal that off-the-shelf fine-tuning techniques are far from adequate to mitigate negative transfer caused by two types of underperforming features in a pre-trained model, including rare features and spuriously correlated features. Rooted in structural causal models of predictions after fine-tuning, we propose a Concept-wise fine-tuning (Concept-Tuning) approach which refines feature representations in the level of patches with each patch encoding a concept. Concept-Tuning minimizes the negative impacts of rare features and spuriously correlated features by (1) maximizing the mutual information between examples in the same category with regard to a slice of rare features (a patch) and (2) applying front-door adjustment via attention neural networks in channels and feature slices (patches). The proposed Concept-Tuning consistently and significantly (by up to 4.76%) improves prior state-of-the-art fine-tuning methods on eleven datasets, diverse pre-training strategies (supervised and self-supervised ones), various network architectures, and sample sizes in a target dataset.

1. Introduction

Pre-trained models, pre-trained by either conventional supervised [26] or resurgent self-supervised strategies [17, 20], undoubtedly constitute a milestone in the artificial intelligence community. Such a resounding success stems from the gap between the heavy reliance of deep neural networks on extensive data on one side and the lack of annotated data in many real-world applications on the other side. Pre-trained models suffice to bridge this gap under the aegis of the well-established fine-tuning paradigm [21].

The practice of fine-tuning, unfortunately but unsurprisingly, is not always outperforming; the notorious problem of

---

*Part of the work was done when the author interned at Tencent AI Lab.
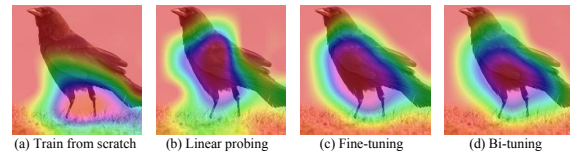†Corresponding Author

Figure 1: Exemplar attentive regions of the model trained (a) from scratch, by (b) linear probing, (c) vanilla fine-tuning, and (d) bi-tuning via Eigen-Grad-CAM [38], where only (a) predicts correctly.

negative transfer [8, 51] arises especially when downstream tasks are out of the distribution of pre-training data. There could be cases where the model trained from scratch outperforms fine-tuning. For example, there exist 3.08% testing images of the downstream dataset CUB [50] on which the model trained from scratch makes correct predictions while fine-tuning the supervised pre-trained model misclassifies. The issue remains even if we resort to one recent state-of-the-art fine-tuning method named Bi-tuning [61], despite offering a smaller percentage of 2.74%. Beyond such a prediction gap between the from-scratch model and fine-tuning models, we do see a gap in attended regions. Fig. 1 shows that the from-scratch model that predicts correctly pays attention to the feet, but fine-tuning models that misclassify concentrate on body-related features, possibly mislead by ImageNet pre-trained models. *The devil that accounts for such gaps lies in those underperforming features offered by the pre-trained model*, among which we focus on the following two types based on our empirical studies.

Type I (*Rare features*)**:** These undertrained features by the pre-training dataset force the attention of the fine-tuned model to be diverted to those well-trained features, although the most discriminative features for the downstream dataset are exactly those undertrained ones.

Fig. 2a demonstrates a slice of rare features, *i.e.*, neck/tail features. Concretely, we mask an original image via Gaussian blurring with two patches preserved, and retrieve its most similar images. The five most similar images retrieved by neck/tail are unfortunately distant (evidenced by colorful necks instead of the expected dark neck) using either pre-trained features (before fine-tuning) or fine-tuned features

(a) Top-5 similar images retrieved gi

Masked image 1
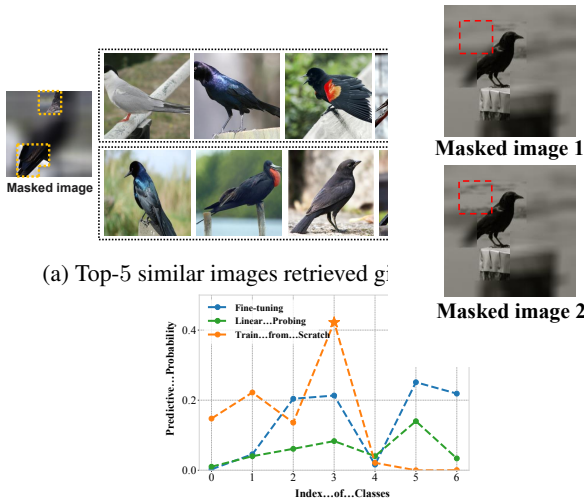
Masked image 2

(b) The union of top-k predictive probability scores by different methods, where the star marks the ground-truth class index 3.

Figure 2: (a) Retrieval of the most similar images based on the cosine similarity between features of the masked image and another one. (b) Given the masked image in (a), the model trained from scratch predicts correctly but both fine-tuning and linear probing fail.

(after fine-tuning), advocating that the fine-grained neck features are undertrained in the coarse-grained pre-training dataset of ImageNet. As a consequence, both linear probing that only fine-tunes the classification head and vanilla fine-tuning lose the attention of the discriminative neck features, resulting in incorrect predictions, while training from scratch where all features are initialized to be unanimously uninformative succeeds.

Type II *(Spuriously correlated features)*: The misleading correlations in the pre-training dataset constitute another source that diverts the attention of the fine-tuned model.

In Fig. 3, we show head features and bird feeder features as a pair of spuriously correlated features. For the masked image 1 with only the two patches of head and tail preserved, all three models make correct predictions; nonetheless, including one more patch describing the bird feeder in the masked image 2 significantly alters the predictions by fine-tuning and linear probing, despite having no influence on that by the from-scratch model. This can be explained by the spurious correlations between head and bird feeder features that exist in another bird class of the pre-training dataset.

This work sets out to develop a fine-tuning strategy that alleviates the negative effects of these two types of under-performing pre-trained features. First, we conclude that maximizing the mutual information between examples in the same class with respect to a particular slice of rare features is contributory to draw the attention to refining rare features to be discriminative. Second, by investigating the causal models of predictions after fine-tuning, we identify that the pre-training dataset as a confounder explains the negative
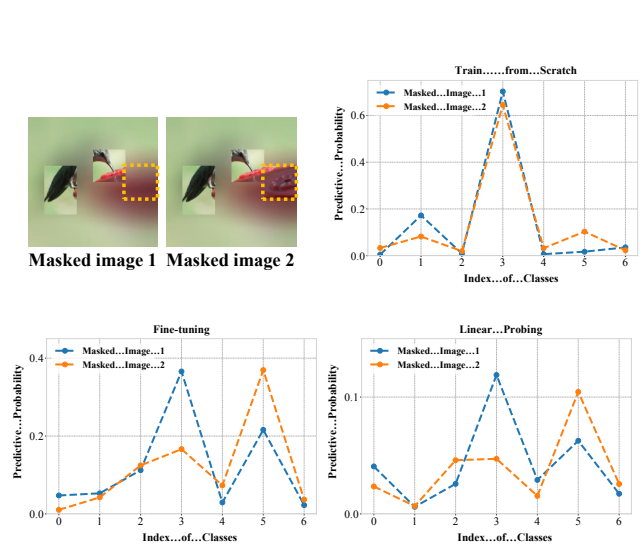

Masked image 1 Masked image 2

Figure 3: An examplar pair of spuriously correlated features, where the model trained from scratch predicts consistently after including one more patch in the masked image 2. Unfortunately, fine-tuning and linear probing predict incorrectly with the introduction of the same patch.

transfer by spuriously correlated features. To this end, we propose to deconfound by the principled front-door adjustment rule. In light of the key role of a patch implementing a slice of features and the fact that each patch usually encrypts a concept, we dub the proposed approach "Concept-Tuning".

We summarize our contributions as follows.

- We have identified two specific types of underperforming pre-trained features that give rise to negative transfer and revealed the root cause of their negative impacts, upon which more principled fine-tuning techniques can be developed.
- To our best knowledge, Concept-Tuning is the first that fine-tunes concept-level sliced features, theoretically analyzed to offer high invariance to the confounder of the pre-training dataset.
- Concept-Tuning improves the state-of-the-art sample-level fine-tuning methods by a large margin (up to 4.76%) and with consistency in eight classification datasets, seven pre-trained models (supervised and self-supervised), three network architectures, and different sample sizes in downstream tasks. Moreover, Concept-Tuning can extend well to semantic segmentation and domain generalization tasks.

2. Related Work

Fine-tuning   The majority of existing fine-tuning methods focus on better exploiting the knowledge of a pre-trained model from different perspectives. For example, L2SP [31] regularizes weights of the fine-tuned model to be close to those of the pre-trained model by imposing the ℓ_2 constraint,

refraining the network from forgetting useful knowledge. REGSL [30] further introduces a layer-wise parameter regularization, where the constraint strength gradually reduces from the top to bottom layers. DELTA [32] imposes constraints on generated activate feature maps instead of weights, where the constraint on each channel is weighted by the channel importance. BSS [9] penalizes smaller singular values of learned feature representations to suppress negative transfer of spectral components. StochNorm [27] regularizes the moving statistics in batch normalization layers to mitigate over-fitting. Co-tuning [58] proposes to explore the information encoded in the last task-specific layers of a pre-trained model, which is usually disregarded, via learning the relationship between the categories of the pre-training dataset and those of a downstream dataset. Apart from these regularization methods, Bi-tuning [61] , Core-tuning [59] and COIN [40] introduce supervised contrastive learning [20, 24] to better leverage the label information in the target dataset with more discriminative representations as a result. However, these methods use image-wise contrastive loss, which is orthogonal to our proposed concept-wise losses. In addition to the above methods that fine-tune all the parameters, some recent methods transfer a large-scale pre-trained model (*e.g.*, ViT [13]) to downstream tasks by freezing the pre-trained backbone while tuning only a small number of newly introduced trainable parameters and the classification head. For example, VPT [23] inserts a few learnable parameters (prompts) in input spaces of the first few layers, and SSF [33] introduces a small number of learnable parameters for feature shifting after each linear layer; however, neither of them targets negative transfer specifically.

Spurious correlation Most deep neural networks are typically trained to minimize the average loss on a training set, known as empirical risk minimization [48]. Despite their success, they likely suffer from *spurious correlations* which hinder their successful generalization, such as the contextual correlation of co-occurring objects [45], background correlation [53], variant-features correlation [3]. The key to solving this problem is to tell invariant feature representations apart from variant ones in the training dataset, so as to enforce the model to rely on the invariant representations only [34, 37, 35]. Recently, researchers also leverage the pre-trained models to address spurious correlations in a downstream dataset [25, 47]. Different from them, we focus on alleviating negative transfer from a pre-trained model with spurious correlated features to a downstream dataset.

Patch representation The majority of existing image classification studies learn the feature representation of an entire image globally, which inevitably lose position-aware information due to the average or max-pooling operation. There has been a line of literature that learns patch-level features

to deal with this problem. For example, Xiao et al. [54] proposed an approach that learns better feature representations of regions by maximizing the similarity in convolution features of the corresponding regions between two augmented views. Xie et al. [55] imposed the contrastive loss between a global image and its local patches to align global and local feature representations. Xie et al. [56] first detected the foreground object regions, based on which they implemented contrastive learning of these regions to specifically improve feature representations of objects. How to embrace these ideas of patch-level representation learning mainly targeting object detection towards better fine-tuning of pre-trained models remains an open but intriguing question.

3. Preliminary

Before proceeding to the proposed fine-tuning approach, we first introduce the notations that we use. Suppose that we are provided with a dataset $\mathcal{D} = \mathcal{D}^{tr} \cup \mathcal{D}^{te}$, where the training set $\mathcal{D}^{tr} = \{\mathbf{x}_i^{tr}, y_i^{tr}\}_{i=1}^{n_i^{tr}}$ consists of n_i^{tr} training examples and the testing set $\mathcal{D}^{te} = \{\mathbf{x}_i^{te}, y_i^{te}\}_{i=n_i^{tr}+1}^{n_i^{tr}+n_i^{te}}$ is made up of n_i^{te} testing examples. $\mathbf{x}_i^{tr}$ ($\mathbf{x}_i^{te}$) and y_i^{tr} (y_i^{te}) denote the features and label of the i-th training (testing) example, respectively. Note that the labels for the testing set are used solely for evaluation purposes. Besides, we have access to a pre-trained model $f_\theta(\cdot)$ pre-trained on a dataset $\mathcal{D}^p$ as an initialization feature extractor. In this paper, we aim to learn a classifier $W(\cdot)$ and fine-tune the feature extractor $f_\theta(\cdot)$ through the training set $\mathcal{D}^{tr}$, so that the function $W(f_\theta(\mathbf{x})) \mapsto y$ predicts well on the testing set $\mathcal{D}^{te}$.

Supervised contrastive fine-tuning Besides the common cross entropy loss $\mathcal{L}_{CE}(W(f_\theta(\mathbf{x})), y)$ adopted in vanilla fine-tuning, recent state-of-the-art fine-tuning methods [59, 61] introduce the supervised contrastive loss [24] for better tuning of feature representations, *i.e.*,

$$\mathcal{L}_{con} = -\frac{1}{n} \sum_{i=1}^{n} \frac{1}{|\mathcal{P}_i|} \sum_{k^+ \in \mathcal{P}_i} \log \frac{\exp(q_i \cdot k^+/\tau)}{\sum_{k^- \in \mathcal{A}_i} \exp(q_i \cdot k^-/\tau)}.$$

Here $\mathcal{P}_i$ and $\mathcal{A}_i$ denote the set of positive representations from the same class to pull towards the query representation q_i as an anchor, and the set of negative representations from different classes to push away, respectively; k^+ and k^- denote a sample from the positive representation set $\mathcal{P}_i$ and the negative representation set $\mathcal{A}_i$, respectively; τ is a temperature hyper-parameter. Specifically, the representation space for application of this supervised contrastive loss could be either $q_i = W(f_\theta(\mathbf{x}_i^{tr}))$ after the classifier [61] or $q_i = f_\theta(\mathbf{x}_i^{tr})$ after the feature extractor [59]. In our work, we also follow the best practice by including $\mathcal{L}_{con}$ and taking $q_i = W(f_\theta(\mathbf{x}_i^{tr}))$, which constitutes the first part of our
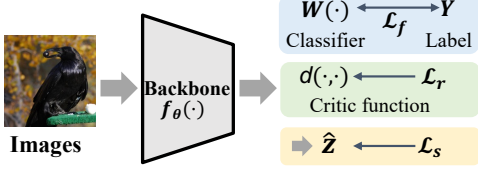
Figure 4: Overview of our proposed approach.

(a) Structural causal model of pre-dictions   (b) Front-door adjustment

Figure 5: A causal probe into the negative impact of spuriously correlated features in pre-trained models.

training objective in Fig. 4,

$$\mathcal{L}_f = \mathcal{L}_{CE}(W(f_\theta(\mathbf{x})), y) + \mathcal{L}_{con}. \tag{1}$$

4. Proposed Approach

In Section 1, we illustrated two types of underperforming features in a pre-trained model that lead fine-tuning astray: rare features and spuriously correlated features. To mitigate such negative transfer, we first investigate the roles of these features in impairing downstream predictions. Based on this investigation, we propose our principled approach. As shown in Fig. 4, our approach includes $\mathcal{L}_r$, which improves the learning of rare features, and $\mathcal{L}_s$, which lessens the influence of spuriously correlated features.

Rare features. Given a training example $\mathbf{x}_i^{tr}$, we denote its feature representation by the pre-trained model as $F_i = f_\theta(\mathbf{x}_i^{tr})$. According to what we have established in Section 1, a rare feature F_i^r is a slice of F_i that should play a greater role than the other slices in classification while being not discriminative, *i.e.*, $p(y_i^{tr}|F_i^r) \approx p(y'|F_i^r)$. $y' \neq y_i^{tr}$ here denotes the label of a different category from y_i^{tr}. Imposing the example-level objective $\mathcal{L}_f$ only without focusing on slices of rare features, similar to vanilla fine-tuning, gets stuck again with those well-trained but nonessential features.
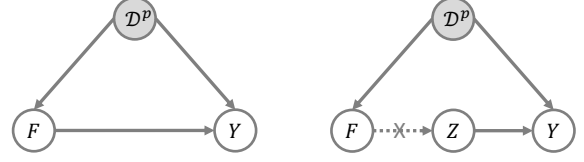
To improve the discriminative ability of rare features pointedly, we propose to maximize the mutual information between examples in the same class with respect to any r-th slice of rare features (*e.g.*, neck features in Section 1), *i.e.*,

$$\max \ I(F_i^r; F_j^r), \quad s.t. \quad y_i^{tr} = y_j^{tr}, \tag{2}$$

where $I(\cdot; \cdot)$ denotes the mutual information. By zooming into each slice feature of an example, this mutual information favourably prioritizes rare features and makes them more discriminative as expected. Despite the notorious challenge of estimating the mutual information, connections have been made of the contrastive loss to maximization of mutual information in [39, 46], showing that

$$I(F_i^r; F_j^r) \geq \log K^- - \mathcal{L}_r, \tag{3}$$

where K^- is the number of negative examples from other classes. Instead, we resort to minimize the contrastive loss

$$\mathcal{L}_r = -\mathbb{E}_{\{F_i^r\}_{i=1}^{K^+}, \{F_j^r\}_{j=1}^{K^+}, \{F_k^r\}_{k=1}^{K^-}} [\log \frac{d(F_i^r, F_j^r)}{\sum_{k=1}^{K^-} d(F_i^r, F_k^r)}],$$

where K^+ is the number of positive examples in the same class as in supervised contrastive loss [24] and $y_i^{tr} \neq y_k^{tr}$.

The challenge now reduces to defining the critic function $d(\cdot, \cdot)$ which is expected to evaluate the similarity between the same r-th slice of rare features between different examples. Take the neck features in Section 1 as an example again. We seek for $d(F_i^r, F_j^r)$ that pulls the neck features of the i-th image (characterizing F_i^r) and those of the j-th (describing F_j^r) given a bird class, say Green Violetear. Fortunately, the earth mover's distance (EMD) [2] function that automatically searches the most similar slice across examples meets this requirement. Therefore, we are inspired to implement

$$d(F_i^r, F_j^r) = \exp(\frac{F_i^r \cdot \xi^*(F_i^r)}{\|F_i^r\| \cdot \|\xi^*(F_i^r)\|} \cdot \frac{1}{\tau}), \tag{4}$$

where $\xi^* = \arg\min_{\xi:\{F_i^{r'}\}_{r'=1}^R \to \{F_j^{r'}\}_{r'=1}^R} \sum_{r'} \|F_i^{r'} - \xi(F_i^{r'})\|$ represents the optimal matching flow among all one-to-one mappings $\xi \in \mathbb{R}^{R \times R}$. $\xi^*(F_i^r)$ hopefully returns the slice of features from the j-th example in the same semantic slice, e.g., neck features, as the i-th example. Consequently, we have the empirical contrastive loss used as,

$$\mathcal{L}_r = -\frac{1}{n} \sum_{i=1}^{n} \frac{1}{K^+} \sum_{j \in \{y_i^{tr} = y_j^{tr}\}}^{K^+} \log \frac{d(F_i^r, F_j^r)}{\sum_{k \in \{y_i^{tr} \neq y_k^{tr}\}}^{K^-} d(F_i^r, F_k^r)}. \tag{5}$$

We leave the details of how to define a patch feature as a slice of features F_i^r as well as more details on the implementation of this contrastive loss in Section 5.1.

Spuriously correlated features Despite the ground-truth label y_i^{tr} (*e.g.*, carriage), the spuriously correlated features (*e.g.*, describing a person) are correlated with another label $y' \neq y_i^{tr}$ (*e.g.*, horse) more frequently in the pre-training dataset $\mathcal{D}^p$, thereby likely misguiding the prediction of y', *i.e.*, $p(y_i^{tr}|F) \ll p(y'|F)$ as demonstrated in Section 1. To investigate more closely the misleading role of the spuriously

correlated features, we construct the structural causal model as shown in Fig. 5a. The causal link $\mathcal{D}^p \rightarrow F \rightarrow Y$ denotes that the features F extracted by the pre-trained model on the pre-training dataset $\mathcal{D}^p$ will be fine-tuned to predict the label Y via the classifier. Unfortunately, besides this legitimate causal path that the model is expected to learn, the "back-door" path $F \leftarrow \mathcal{D}^p \rightarrow Y$ gives rise to a spurious correlation from F to Y, making $\mathcal{D}^p$ unfavorably a confounder [42].

To this end, we propose to deconfound and thereby alleviate the negative impact of spuriously correlated features. There are two deconfounding techniques [41, 42]: back-door adjustment, which disconnects the causal link from $\mathcal{D}^p$ to F, and front-door adjustment, which introduces a mediator Z from F to Y while severing the link from F to Z [1]. We adopt the front-door adjustment approach since back-door adjustment requires access to the pre-training dataset $\mathcal{D}^p$, which is always unavailable in our problem. Concretely, as shown in Fig. 5b, the front-door adjustment estimates the true causal effect between F and Y by deconfounding the confounder $\mathcal{D}^p$, *i.e.*,

$$
\begin{aligned}
p(Y|do(F)) &= \sum_z p(Z=z|F)p(Y|Z=z)) \\
&= \sum_z p(Z=z|F)\sum_f p(F=f)[p(Y|F=f,Z=z)] \\
&= \mathbb{E}_{Z|F}\mathbb{E}_F[p(Y|F=f,Z=z)] \\
&= \mathbb{E}_{Z|F}\mathbb{E}_F \text{Softmax}(\varphi(F,Z)) \\
&\approx \text{Softmax}[\varphi(\mathbb{E}_F F, \mathbb{E}_{Z|F} Z))] \\
&= \text{Softmax}\big[\varphi\big(\sum_f p(F=f)\mathbf{f}, \sum_z p(Z=z|F)\mathbf{z}\big)\big],
\end{aligned}
$$
(6)
(7)

where $do(\cdot)$ is the do-operation in causal inference, z represents a slice of features from F. f is a stratum of F and φ is a classifier. $\mathbf{z}$ and $\mathbf{f}$ are specific embedding vectors that correspond to the two variables z and f, respectively. The fourth equation comes from the implementation of $p(Y|F=f,Z=z)$ as the classifier φ with a Softmax layer. The approximation is derived from the normalized weighted geometric mean (NWGM) [57], which moves the outer expectation in Eqn. (6) inside the Softmax function and the classifier φ. More details on the derivation of Eqn. (7) can be found in Appendix A.1.

To be compatible with the slice-level break-down of F into patches when dealing with rare features, here we formulate Z as sets of patches (or concepts) specifically. As a result, the expectations $\mathbb{E}_F F = \sum_f p(F=f|F)\mathbf{f}$ and $\mathbb{E}_{Z|F} Z = \sum_z p(Z=z|F)\mathbf{z}$ in Eqn. (7) can be easily actualized with attention neural networks [49, 16] in the level of channels and patches, respectively. We leave more implementation details of the attentional neural networks in Section 5.1.

The approximation error in Eqn. (7) makes the causal

---

link from F to Z not fully disconnected, thereby leaving the confounder $\mathcal{D}^p$ not fully deconfounded. On this account, we propose to further minimize the correlation between F and the feature input $\hat{Z}$ which is measured by their mutual information $I(\hat{Z};F)$. Inspired by variational information bottleneck [1] where minimizing the mutual information reduces to minimizing the Kullback-Leibler (KL) divergence between $p(\hat{Z}|F)$ and a standard Gaussian distribution $\mathcal{N}(0,1)$, we model $p(\hat{Z}|F) = \mathcal{N}(\mu^{\hat{Z}}, \sigma^{\hat{Z}})$ and minimize the KL loss $\text{KL}[\mathcal{N}(\mu^{\hat{Z}}, \sigma^{\hat{Z}}), \mathcal{N}(0,1)]$ instead. Specifically, we generate the mean $\mu^{\hat{Z}}$ and variance $\sigma^{\hat{Z}}$ by projecting the concatenation of $\sum_f p(F=f)\mathbf{f}$ and $\sum_z p(Z=z|F)\mathbf{z}$.

Combining the deconfounded prediction $p(Y|do(F))$ and the minimization of $I(\hat{Z};F)$, we conclude the following loss that mitigates the spurious correlations caused by $\mathcal{D}^p$ as,

$$
\mathcal{L}_s = \mathcal{L}_{CE}(p(Y|do(F)), y) + \text{KL}[\mathcal{N}(\mu^{\hat{Z}}, \sigma^{\hat{Z}}), \mathcal{N}(0,1)].
$$
(8)

Proposition 4.1. *Given that $\hat{Z}$ depends on $\mathcal{D}^p$ only through F, i.e., $\mathcal{D}^p \rightarrow F \rightarrow \hat{Z}$, we have*

$$
I(\hat{Z};\mathcal{D}^p) \leq I(\hat{Z};F) - I(F;Y)
$$
(9)

and minimizing $I(\hat{Z};F)$ leads to invariant representation $\hat{Z}$ that is maximally insensitive to the confounder $\mathcal{D}^p$.

This proposition justifies that retaining the minimum information $I(\hat{Z};F)$ through either deconfounding or minimizing $\text{KL}[\mathcal{N}(\mu^{\hat{Z}}, \sigma^{\hat{Z}}), \mathcal{N}(0,1)]$ suffices to offset the negative effects of the spuriously correlated features during fine-tuning. We provide the proof for Proposition 4.1 in Appendix A.2.

Taking all the loss functions that we have introduced for supervised contrastive fine-tuning in Eqn. (1), improving rare features in Eqn. (5), and alleviating spuriously correlated features in Eqn. (8), we finally obtain the overall objective function that achieves concept-level fine-tuning as follows,

$$
\mathcal{L} = \mathcal{L}_f + \alpha\mathcal{L}_r + \beta\mathcal{L}_s
$$
(10)

where α and β are the balancing hyper-parameters that governs the impact of $\mathcal{L}_r$ and $\mathcal{L}_s$. The pseudo-code of the whole algorithm is provided in Appendix C.

5. Experiments

To evaluate the effectiveness of Concept-Tuning, we conduct extensive experiments to answer the following questions: **Q1:** How do our methods perform compared to state-of-the-art fine-tuning methods? **Q2:** Can the proposed Concept-Tuning consistently improve the performance under different data sizes? **Q3:** Can our methods be applied to different pre-trained models and architectures? **Q4:** Do our methods indeed alleviate the negative impact of the two types of underperforming pre-trained features?

5.1. Experiment setup

Datasets. We evaluate our methods on eight image classification datasets, covering a wide range of fields: CUB-200-2011 [50], Stanford Cars [28], FGVC Aircraft [36], CIFAR10, CIFAR100 [29], Vegetable [44], ISIC [11] and Caltech101 [15]. Among them, ISIC is a medical dataset that targets classifying skin lesion images into seven possible disease categories. We split the datasets as per previous works [58, 61] to make fair comparison. Further, we apply Concept-Tuning to semantic segmentation tasks on PASCAL VOC [14] and ADE20k [63], and domain generalization task on DomainNet [43]. We provide more descriptions for all datasets (*e.g.*, the statics of each dataset and the train/test split) in Appendix E.1.

Baselines. We compare Concept-Tuning with recent advanced fine-tuning methods, which can be roughly categorized into three classes: regularization-based methods, supervised contrastive methods, and parameter-efficient fine-tuning methods. (1) Regularization-based methods including L2SP [31] and REGSL [30] which regularize the weights of the fine-tuned model, DELTA [32] and BSS [9] which impose constraints on the generated feature maps, and Co-tuning [58] which explores label information of the pre-trained dataset as an additional regularization on the downstream predictions. (2) Supervised contrastive methods including Bi-tuning [61], Core-tuning [59] and COIN [40], all of which introduce supervised contrastive learning into fine-tuning and benefit from more discriminative feature representations by pulling features from the same class together and simultaneously pushing apart features from different classes in the embedding space. Specifically, COIN [40] fine-tunes a model with L_{con} in the first several epochs and later the joint of L_{CE} and L_{con}. (3) Parameter-efficient fine-tuning methods including VPT [23] and SSF [33], which freeze the pre-trained backbone while tuning only a small number of newly introduced trainable parameters and the classification head. Since the last line of methods targets large-scale pre-trained models, we include them in only the experiments involving ViT-B/16. In addition, we include vanilla fine-tuning with only the cross-entropy loss as a baseline.

Models. Following [61], we mainly use ResNet-50 pre-trained on ImageNet-1k in a supervised manner to evaluate different fine-tuning methods on various datasets. In addition, we conduct experiments with ResNet-50 that are pre-trained by four other self-supervised pre-training strategies (*i.e.*, MoCo-V2 [7], SimCLR [6], SwAV [4] and BYOL [17]) to evaluate the effectiveness of Concept-Tuning on different pre-trained models. Moreover, we follow VPT [23] to evaluate the compatibility of the proposed method with larger models: ResNet-101 and ViT-B/16 [13]), where ResNet-101 is supervised pre-trained on ImageNet-1K

and ViT-B/16 is pre-trained on ImageNet-21K[2].

Implementation details. We implement our methods based on the Transfer Learning Library[3]. For the baselines, we use the same training strategies and default hyper-parameters settings as in the previous works [61, 59]. During training, we utilize a standard augmentation strategy by performing random resize-crop to 224×224, random horizontal flip, and normalization with ImageNet means and deviations. Following MoCo [7], we keep a momentum-updated model and maintain two groups of queues of features. Specifically, denoting the tuning model as θ_q and the momentum-updated model as θ_k, we update θ_k by $\theta_k \leftarrow m\theta_k + (1-m)\theta_q$ where m as the momentum coefficient is set as 0.999 by default. Each queue in the first group of queues stores n ℓ_2 normalized feature representations for examples in each class and each queue in the second group stores n ℓ_2 normalized patch representations for each class, where the representations are generated by the momentum-updated model and n is a hyper-parameter controlling the size of a queue. Moreover, we set the temperature $\tau = 0.07$ [52]. More details of the hyper-parameter settings tuned by validation sets are provided in Appendix E.2. All results are averaged over three trials.

Implementation details in rare features. To define a patch feature as a slice of features F^r, we first determine the regions of rare features and then extract features of the cropped regions. In detail, we obtain the most confusing class based on the classification logits, upon which we retrieve the classification weights of the ground truth class $W_y \in \mathbb{R}^C$ (C represents the number of channels in F) and the confusing class $W_{y'}$, respectively. Note that the input features F to the classification head are all non-negative thanks to the ReLU operation so that $u = W_y - W_{y'}$ demonstrates each channel's importance in classifying these two classes. u_i close to zero indicates that the i-th channel feature is not that discriminate. Inspired by previous works [62, 60] that connect each channel of the convolutional feature map to a visual concept, we choose the attentive regions of the channels whose u are closest to zero as regions of the rare features. In our experiments, we use 9 channels with u closest to zero and thereby crop 9 patches in size of 64×64, each of which is at the most attentive position of a selected channel.

Implementation details in spurious correlated features. Given a feature representation $F \in \mathbb{R}^{C \times H \times W}$ of an example, where C represents the channel, H and W represent the height and width of the feature, we feed it into attention networks to get two expectations $\mathbb{E}_{Z|F}Z$ and $\mathbb{E}_F F$. $\mathbb{E}_{Z|F}Z$ via a patch attention module: we first obtain two new features $\{K, Q\} \in \mathbb{R}^{C/8 \times H \times W}$ to better model the probability $p(Z = z|F)$ in the embedded space as well as reduce the computation complexity through two 1×1 convolution layers, respectively. Then both K and Q are reshaped to

---

[2]https://github.com/rwightman/pytorch-image-models
[3]https://github.com/thuml/Transfer-Learning-Library

Table 1: Top-1 accuracy (%) on various datasets using supervised pre-trained ResNet-50.

Method	CUB	Cars	Aircraft	CIFAR10	CIFAR100	Vegetable	ISIC	Caltech101	Avg.
Vanilla Fine-tuning	78.01 ± 0.16	87.20 ± 0.19	81.13 ± 0.21	96.33 ± 0.12	83.56 ± 0.17	88.10 ± 0.16	86.07 ± 0.24	92.20 ± 0.24	86.56
L2SP	78.44 ± 0.17	86.58 ± 0.26	80.98 ± 0.29	96.29 ± 0.23	83.01 ± 0.19	88.04 ± 0.21	85.99 ± 0.22	92.36 ± 0.17	86.46
DELTA	78.63 ± 0.18	86.32 ± 0.20	80.44 ± 0.20	93.77 ± 0.16	78.98 ± 0.24	88.21 ± 0.17	86.05 ± 0.19	91.95 ± 0.33	85.54
BSS	78.85 ± 0.31	87.63 ± 0.27	81.48 ± 0.18	96.35 ± 0.31	83.80 ± 0.15	88.60 ± 0.26	85.55 ± 0.17	92.51 ± 0.15	86.85
Co-tuning	81.24 ± 0.14	89.53 ± 0.09	83.87 ± 0.09	96.42 ± 0.26	81.40 ± 0.22	88.26 ± 0.19	85.21 ± 0.13	92.76 ± 0.18	87.32
REGSL	81.60 ± 0.21	88.83 ± 0.18	84.07 ± 0.23	97.16 ± 0.17	83.96 ± 0.20	87.43 ± 0.25	86.09 ± 0.19	92.21 ± 0.22	87.67
Bi-tuning	82.93 ± 0.23	88.47 ± 0.11	84.01 ± 0.33	96.80 ± 0.20	84.44 ± 0.18	89.60 ± 0.26	86.23 ± 0.28	92.67 ± 0.08	88.14
Core-tuning	81.99 ± 0.12	91.68 ± 0.16	86.71 ± 0.15	97.28 ± 0.14	$\mathbf{86.09 \pm 0.21}$	88.63 ± 0.25	85.13 ± 0.16	91.66 ± 0.11	88.65
COIN	82.76 ± 0.19	89.88 ± 0.21	87.88 ± 0.28	97.43 ± 0.25	85.49 ± 0.22	88.52 ± 0.15	86.61 ± 0.24	92.67 ± 0.18	88.90
Ours 1 ($+\mathcal{L}_r$)	84.86 ± 0.26	92.66 ± 0.28	88.84 ± 0.25	$\mathbf{97.54 \pm 0.12}$	85.96 ± 0.09	89.79 ± 0.13	$\mathbf{86.63 \pm 0.21}$	92.97 ± 0.14	89.91
Ours 2 ($+\mathcal{L}_r+\mathcal{L}_s$)	$\mathbf{85.02 \pm 0.21}$	$\mathbf{92.90 \pm 0.24}$	$\mathbf{89.65 \pm 0.30}$	97.52 ± 0.14	85.59 ± 0.15	$\mathbf{90.10 \pm 0.18}$	86.01 ± 0.26	$\mathbf{93.15 \pm 0.10}$	$\mathbf{89.99}$

$\{K_R, Q_R\} \in \mathbb{R}^{C/8 \times N}$, where $N = H \times W$ is the number of patch features. After that, we obtain the patch-wise attention through $P = Softmax(Q_R^T K_R) \in \mathbb{R}^{N \times N}$, where T indicates transpose operation. Meanwhile, we also obtain a new feature map $V \in \mathbb{R}^{C \times H \times W}$ through a 1×1 convolution layer and then reshape it to $V_R \in \mathbb{R}^{C \times N}$. Lastly, we get the selected feature in patches via a matrix multiplication operation between P and V: $V_R P \in \mathbb{R}^{C \times N}$. $\mathbb{E}_F F$ via a channel attention module: here we directly calculate the channel attention map S from F. Specifically, we first reshape F into $F_R \in \mathbb{R}^{C \times N}$ and then obtain $S = F_R F_R^T \in \mathbb{R}^{C \times C}$ to model $p(F = f | F)$, and then $\mathbb{E}_F F = \sum_f p(F = f | F)\mathbf{f}$ can be calculated through a matrix multiplication operation between F_R and S: $S F_R \in \mathbb{R}^{C \times N}$. Finally, the two expectations are aggregated by element-wise sum operation and then average pooling along dimension N. Please kindly refer to Appendix C for the pseudo-code and Appendix D for the structure of the attention network.

5.2. Experimental results

Comparison with previous methods. We first report the performances of different methods on eight image classification datasets using ResNet-50 [22] supervised pre-trained on ImageNet-1k in Table 1. Concept-Tuning outperforms all competitors by a large margin (average 1.34% improve-

Table 2: Top-1 accuracy (%) on three datasets under different sampling rates using supervised pre-trained ResNet-50.

Dataset	Method	Sampling Rates				
		15%	30%	50%	100%	Avg.
CUB	Vanilla fine-tuning	45.25 ± 0.12	59.68 ± 0.21	70.12 ± 0.29	78.01 ± 0.16	63.27
	Bi-Tuning	55.83 ± 0.04	69.52 ± 0.24	77.17 ± 0.13	82.93 ± 0.23	71.36
	Core-Tuning	55.94 ± 0.07	68.54 ± 0.16	76.41 ± 0.18	81.99 ± 0.12	70.72
	Ours 1	58.15 ± 0.11	71.16 ± 0.18	77.98 ± 0.20	84.86 ± 0.26	73.04
	Ours 2	$\mathbf{60.70 \pm 0.25}$	$\mathbf{73.18 \pm 0.22}$	$\mathbf{78.68 \pm 0.14}$	$\mathbf{85.02 \pm 0.21}$	$\mathbf{74.40}$
Cars	Vanilla fine-tuning	36.77 ± 0.12	60.63 ± 0.18	75.10 ± 0.21	87.20 ± 0.19	64.93
	Bi-Tuning	48.86 ± 0.22	73.05 ± 0.29	81.10 ± 0.07	88.47 ± 0.11	72.87
	Core-Tuning	53.79 ± 0.17	77.27 ± 0.24	85.56 ± 0.18	91.68 ± 0.16	77.08
	Ours 1	55.00 ± 0.28	78.57 ± 0.27	86.63 ± 0.15	92.66 ± 0.28	78.22
	Ours 2	$\mathbf{57.32 \pm 0.19}$	$\mathbf{79.99 \pm 0.22}$	$\mathbf{87.60 \pm 0.12}$	$\mathbf{92.90 \pm 0.24}$	$\mathbf{79.45}$
Aircraft	Vanilla fine-tuning	39.57 ± 0.20	57.46 ± 0.12	67.93 ± 0.28	81.13 ± 0.21	61.52
	Bi-Tuning	47.91 ± 0.32	64.45 ± 0.23	72.40 ± 0.22	84.01 ± 0.33	67.19
	Core-Tuning	$\mathbf{50.38 \pm 0.34}$	68.78 ± 0.28	77.86 ± 0.26	86.71 ± 0.15	70.93
	Ours 1	49.02 ± 0.28	68.05 ± 0.17	78.04 ± 0.16	88.84 ± 0.25	70.99
	Ours 2	50.05 ± 0.23	$\mathbf{69.10 \pm 0.27}$	$\mathbf{78.40 \pm 0.24}$	$\mathbf{89.65 \pm 0.30}$	$\mathbf{71.80}$

ments on eight datasets), especially on fine-grained datasets (*e.g.*, 2.09% improvements on CUB and 2.94% on Aircraft compared to the previous best method). Strictly regularizing the weights or features (*e.g.*, L2SP and DELTA) between the pre-trained and fine-tuned models could exacerbate negative transfer and thus achieve poor performances, which is in most cases even worse than vanilla fine-tuning. Relaxing the weights-regularization on different layers as REGSL or only penalizing smaller singular values of features as BSS could alleviate the forehead problem and achieve better performances. Moreover, introducing supervised contrastive learning during fine-tuning(*e.g.*, Bi-tuning and Core-tuning) can achieve superior performances as it efficiently leverages the label information via supervised contrastive learning. We attribute less improvement on CIFAR to the low resolution (32×32) images, which makes accurate extraction of concept-level features challenging. However, this does not limit the applicability of Concept-Tuning in light of prevailing high-resolution images nowadays. As shown in the averaged accuracy on eight datasets, our method achieves the best performance by successfully reducing the negative transfer caused by the two underperforming features in the pre-trained model.

Results with different data sizes. To verify the performance of our methods on different sizes of training data, especially on a small training dataset, we follow [58] to sample 15%, 30%, 50% and 100% of data for training. We show the performances of the proposed method and a part of the baselines in Table 2 and the full table in Appendix F. We find that our method consistently surpasses all previous methods under different sampling rates. This indicates that even when the training data is limited, our methods can resolve the severe negative transfer in the pre-training model and achieve much better performance than the previous methods. For example, under 15% sampling rate on CUB, our method is 4.87% and 4.76% higher than Bi-tuning and Core-tuning, respectively.

Results with different unsupervised pre-training strategies. To evaluate the effectiveness of our methods on unsupervised pre-trained models, we further conduct extensive experiments on ResNet-50 pre-trained by four widely-used self-

supervised pre-training methods MoCo-V2 [7], SimCLR [6], SwAV [4] and BYOL [17] and ViT-B/16 pre-trained by two self-supervised pre-training methods (MAE [19] and MoCo-v3 [10]). All the models are pre-trained on ImageNet and the pre-trained weights are provided by their authors. As shown in Table 3, our methods work well under different self-supervised pre-trained methods, showing that the effectiveness of Concept-Tuning is not bounded to specific pre-training models. More results (*e.g.*, MAE [19] and MoCo-v3 [10])) are available in Appendix F.

Results with different backbone architectures. Previously we conduct experiments on ResNet-50, while the effectiveness of Concept-Tuning on more architectures is unexplored. Here, we conduct experiments on three different backbone architectures: ResNet-50, ResNet-101 [22], which are supervised pre-trained on ImageNet-1k, and Vit-B/16 [13] supervised pre-trained on ImageNet-21k. As shown in Table 4, Concept-Tuning performs well on all three architectures. Despite the superior performances of vanilla fine-tuning on ViT-B/16, Concept-Tuning obtains non-trivial performance gains (*e.g.*, 2.6% on CUB), verifying the effectiveness of Concept-Tuning on large pre-trained models.

5.3. Ablation studies

The effect of the two losses $\mathcal{L}_r$ and $\mathcal{L}_s$. We first conduct ablation studies of Concept-Tuning on the loss regarding rare features $\mathcal{L}_r$, spuriously correlated features $\mathcal{L}_s$, and invariant representation KL. As shown in Table 5, only resolving rare features via $\mathcal{L}_r$ can improve the performances by a large margin, indicating that it leads to better feature representations and thus achieves better classification results, and only resolving spuriously correlated features to disconnect causal links via $\mathcal{L}_s$ can also obtain obvious improvements. Further, minimizing $I(\hat{Z}, F)$ through the proposed KL loss promotes invariant representations $\hat{Z}$, verified by the improvements on L_s beyond L_s w/o KL. Moreover, we find that simultaneously applying $\mathcal{L}_r$ and $\mathcal{L}_s$ yields the best results, as shown in the last row of Table 5.

More ablation studies (*i.e.*, the trade-off weight α, the

Table 3: Top-1 accuracy (%) on three datasets using four different pre-trained ResNet-50.

Dataset	Method	Pre-trained method				
		MoCo-V2	SimCLR	SwAV	BYOL	Avg.
	Vanilla fine-tuning	76.72 ± 0.21	76.51 ± 0.28	80.45 ± 0.32	81.29 ± 0.29	78.74
	Bi-tuning	79.48 ± 0.24	75.73 ± 0.25	81.72 ± 0.23	82.02 ± 0.29	79.74
CUB	Core-tuning	77.93 ± 0.18	77.55 ± 0.15	80.60 ± 0.27	78.46 ± 0.18	78.64
	Ours 1	82.48 ± 0.14	78.18 ± 0.20	83.47 ± 0.22	83.38 ± 0.18	81.88
	Ours 2	$\mathbf{82.53 \pm 0.21}$	$\mathbf{79.81 \pm 0.23}$	$\mathbf{84.78 \pm 0.32}$	$\mathbf{84.45 \pm 0.29}$	**82.89**
	Vanilla fine-tuning	88.45 ± 0.35	84.53 ± 0.12	88.17 ± 0.21	88.99 ± 0.39	87.54
	Bi-tuning	90.05 ± 0.15	91.75 ± 0.18	90.49 ± 0.27	90.90 ± 0.18	90.80
Cars	Core-tuning	90.87 ± 0.23	91.78 ± 0.26	91.84 ± 0.14	91.95 ± 0.18	91.61
	Ours 1	91.02 ± 0.11	93.27 ± 0.20	93.41 ± 0.26	93.22 ± 0.15	92.73
	Ours 2	$\mathbf{91.75 \pm 0.18}$	$\mathbf{93.36 \pm 0.23}$	$\mathbf{93.79 \pm 0.32}$	$\mathbf{93.68 \pm 0.25}$	**93.15**
	Vanilla fine-tuning	88.60 ± 0.18	87.79 ± 0.24	83.26 ± 0.17	85.03 ± 0.15	86.17
	Bi-Tuning	89.05 ± 0.16	88.69 ± 0.17	85.69 ± 0.13	87.16 ± 0.11	87.65
Aircraft	Core-Tuning	89.02 ± 0.19	89.47 ± 0.21	88.66 ± 0.34	89.74 ± 0.20	89.22
	Ours 1	$\mathbf{89.65 \pm 0.18}$	90.13 ± 0.11	91.42 ± 0.36	90.82 ± 0.22	90.50
	Ours 2	89.32 ± 0.21	$\mathbf{90.85 \pm 0.17}$	$\mathbf{91.75 \pm 0.14}$	$\mathbf{91.21 \pm 0.13}$	**90.76**

Table 4: Top-1 accuracy (%) on two datasets using different architectures with supervised pre-training, where $*$ indicates the results reported in [23] and [33].

Dataset	Method	Architecture			
		ResNet-50	ResNet-101	ViT-B/16	Avg.
	Vanilla fine-tuning	78.01 ± 0.16	82.26 ± 0.34	87.3^*	81.32
	Bi-tuning	82.93 ± 0.23	83.57 ± 0.13	89.23 ± 0.33	85.24
	Core-tuning	81.99 ± 0.12	81.76 ± 0.14	89.30 ± 0.21	84.35
CUB	VPT	$*$	$*$	88.5^*	$*$
	SSF	$*$	$*$	89.5^*	$*$
	Ours 1	84.86 ± 0.26	83.98 ± 0.17	$\mathbf{89.90 \pm 0.18}$	86.25
	Ours 2	$\mathbf{85.02 \pm 0.21}$	$\mathbf{84.12 \pm 0.16}$	89.78 ± 0.25	**86.31**
	Vanilla fine-tuning	87.20 ± 0.19	88.84 ± 0.28	84.5^*	86.85
	Bi-tuning	88.47 ± 0.11	90.82 ± 0.16	92.40 ± 0.19	90.56
	Core-tuning	91.68 ± 0.16	91.49 ± 0.13	91.12 ± 0.34	91.43
Cars	VPT	$*$	$*$	83.6^*	$*$
	SSF	$*$	$*$	89.2^*	$*$
	Ours 1	92.66 ± 0.17	92.63 ± 0.22	92.86 ± 0.19	92.72
	Ours 2	$\mathbf{92.90 \pm 0.24}$	$\mathbf{92.76 \pm 0.24}$	$\mathbf{92.92 \pm 0.26}$	**92.86**

trade-off weight β, the patch size, the temperature τ, and the number of keys stored in the queue) are provided in Appendix F.

5.4. Visualization

To better understand the effectiveness of our methods, we compare the CAMs of different methods, as shown in Fig. 6a. Influenced by the pre-trained model, Fine-tuning and Bi-tuning tend to focus on the reflection in the water. In contrast, our methods could alleviate the problem and attend to regions closer to the model trained from scratch. Moreover, Concept-Tuning focuses on a relatively minor part than Bi-tuning, indicating that our method could effectively resolve those spuriously correlated features. To further verify that our approach can improve the representation of rare features, we test our methods on the masked image in Fig. 2a. As shown in Fig. 6b, Concept-Tuning predicts correctly, while both Fine-tuning and Bi-tuning fail.

To compare the convergence of different methods, we follow COIN [40] to plot the training curves. As shown in Fig. 7, all methods have fully converged to similar training accuracy, while Concept-Tuning achieves the best performance on the testing set.

Table 5: Ablation studies of our methods using ResNet-50 by supervised pre-training, and we report the top-1 accuracy (%) on three datasets. Note that we follow the training strategies in Bi-tuning [61] to implement the experiments with $\mathcal{L}_f$, which is a stronger baseline.

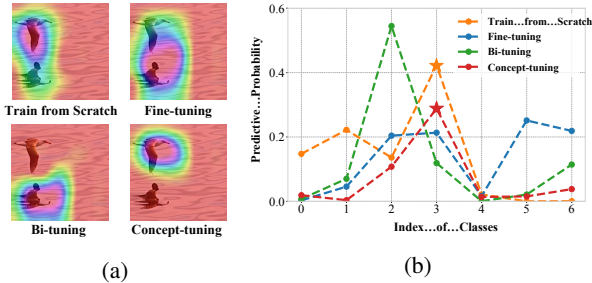| Loss				Dataset		
$\mathcal{L}_f$	$\mathcal{L}_r$	$\mathcal{L}_s$	$\mathcal{L}_s$ w/o KL	CUB	Cars	Aircraft
✓				82.93 ± 0.23	88.47 ± 0.11	84.01 ± 0.33
✓	✓			84.86 ± 0.26	92.66 ± 0.28	88.84 ± 0.25
✓			✓	83.36 ± 0.25	90.49 ± 0.24	86.29 ± 0.31
✓		✓		84.21 ± 0.23	91.84 ± 0.19	87.46 ± 0.28
✓	✓		✓	84.92 ± 0.22	92.55 ± 0.28	89.11 ± 0.36
✓	✓	✓		$\mathbf{85.02 \pm 0.21}$	$\mathbf{92.90 \pm 0.24}$	$\mathbf{89.65 \pm 0.30}$

Figure 6: (a) CAM visualization of 4 methods. Influenced by the pre-trained model, fine-tuning and Bi-tuning will be attracted by the reflection in the water and make wrong predictions, while Concept-Tuning focuses on the bird and predicts correctly. (b) Given a Gaussian-blurred masked image in Fig. 2a, our method alleviates the negative transfer and predicts correctly as the model trained from scratch.
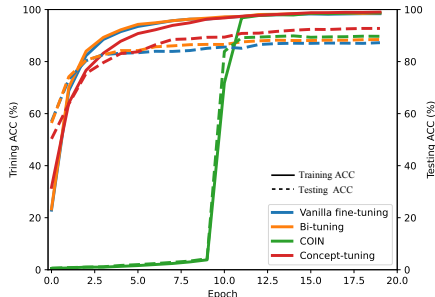

Figure 7: Comparisons of the training curves of different methods on Cars.

5.5. Experiments on semantic segmentation

To explore the feasibility of our methods beyond visual classification, we extend the evaluation of ours to semantic segmentation, where we follow the training protocol in [12] to fine-tune the pre-trained DeepLab-V3 model [5] (ResNet-50 as the backbone) on PASCAL VOC [14] and ADE20k [63], respectively. Following [12], for PASCAL VOC, we train on the VOC2012 training set and its augmented training set [18] with 20k iterations and then test on VOC2012 validation set; for ADE20k, we train on its training set with 80k iterations and test on its validation set. We use SGD with an initial learning rate of 0.01 and a poly decay schedule to train the model. The images are resized to 512×512 for PASCAL VOC and ADE20k, and other hyper-parameters are the same as [12]. We use Mean Intersection over Union (MIoU) to evaluate the performances of different methods. Results in Table 6 corroborate the effectiveness of Concept-Tuning in even segmentation tasks.

Table 6: Results on fine-tuning DeepLab-V3 (ResNet-50) on two semantic segmentation datasets.

Method	PASCAL VOC	ADE20k
Vanilla fine-tuning	76.17	42.42
Core-tuning	76.94	42.97
Ours 2	**77.66**	**43.12**

Table 7: OOD generalization results on DomainNet. Moreover, c/i/p/q/r/s represent different domains; :c means that the model is fine-tuned on domains except for c and evaluated on c.

Method	avg.	:c	:i	:p	:q	:r	:s
Vanilla fine-tuning	45.72	62.98	26.31	52.63	14.20	**67.00**	51.25
Bi-tuning	46.32	64.42	27.65	53.18	14.79	66.36	51.56
Core-tuning	46.06	63.68	26.94	52.89	15.17	65.91	51.78
Ours 2	**46.74**	64.48	28.17	53.68	15.33	66.04	**52.73**

5.6. Benefits to out-of-distribution generalization

We also conduct experiments on the domain generalization task to investigate our method's robustness on out-of-distribution (OOD) datasets. We compare methods on a large-scale multi-source dataset, *i.e.*, DomainNet [43]. DomainNet includes six domains (*i.e.*, Clipart, Infograph, Painting, Quickdraw, Real, and Sketch), and the model is fine-tuned on five domains and evaluated on one remaining domain. Specifically, we use supervised pre-trained ResNet-50 as the backbone. For training, we follow the training protocol in Transfer Learning Library to set the learning rate as 0.01 and the training step as 50, 000. Results in table 7 show that Concept-Tuning surpasses previous fine-tuning methods, advocating its advantage in generalizing to OOD datasets.

6. Conclusions and Discussions

Drew on our preliminary empirical observations, we pinpoint two types of underperforming pre-trained features in pre-trained models that likely give rise to negative transfer, *i.e.*, rare features and spuriously correlated features. In this paper, we develop a highly principled Concept-Tuning, which combats the negative impacts of rare features from an information-theoretic perspective and that of spuriously correlated features based on causal adjustment. By formulating concepts as patches, we have concretely derived the concept-level contrastive loss function and the prediction with attention to channels and patches. Extensive experiments validate the effectiveness of the proposed Concept-Tuning in a handful of downstream tasks with various pre-trained models and under different sample sizes. **Limitations:** we are more than eager to investigate the scope and types of downstream tasks which the proposed Concept-Tuning significantly boosts, which is only partially understood insofar.

References

[1] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *ICLR*, 2017. 5

[2] Alexandr Andoni, Piotr Indyk, and Robert Krauthgamer. Earth mover distance over high-dimensional spaces. In *SODA*, volume 8, pages 343–352, 2008. 4

[3] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019. 3

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 33, 2020. 6, 8

[5] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 9

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020. 6, 8

[7] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 6, 8

[8] Xinyang Chen, Sinan Wang, Bo Fu, Mingsheng Long, and Jianmin Wang. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. *NeurIPS*, 32, 2019. 1

[9] Xinyang Chen, Sinan Wang, Bo Fu, Mingsheng Long, and Jianmin Wang. Catastrophic forgetting meets negative transfer: Batch spectral shrinkage for safe transfer learning. In *NeurIPS*, 2019. 3, 6

[10] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *ICCV*, 2021. 8

[11] Noel Codella, Veronica Rotemberg, Philipp Tschandl, M Emre Celebi, Stephen Dusza, David Gutman, Brian Helba, Aadi Kalloo, Konstantinos Liopyris, Michael Marchetti, et al. Skin lesion analysis toward melanoma detection 2018: A challenge hosted by the international skin imaging collaboration (isic). *arXiv preprint arXiv:1902.03368*, 2019. 6

[12] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 9

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 3, 6, 8

[14] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111:98–136, 2015. 6, 9

[15] Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *CVPRW*, 2004. 6

[16] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, pages 3146–3154, 2019. 5

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 33, 2020. 1, 6, 8

[18] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *IJCV*, pages 991–998. IEEE, 2011. 9

[19] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 8

[20] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 1, 3

[21] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *ICCV*, pages 4918–4927, 2019. 1

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 7, 8

[23] Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. Visual prompt tuning. In *ECCV*. Springer, 2022. 3, 6, 8

[24] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *NeurIPS*, 2020. 3, 4

[25] Polina Kirichenko, Pavel Izmailov, and Andrew Gordon Wilson. Last layer re-training is sufficient for robustness to spurious correlations. *arXiv preprint arXiv:2204.02937*, 2022. 3

[26] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *CVPR*, pages 2661–2671, 2019. 1

[27] Zhi Kou, Kaichao You, Mingsheng Long, and Jianmin Wang. Stochastic normalization. In *NeurIPS*, 2020. 3

[28] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCVW*, pages 554–561, 2013. 6

[29] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[30] Dongyue Li and Hongyang Zhang. Improved regularization and robustness for fine-tuning in neural networks. *NeurIPS*, 2021. 3, 6

[31] Xuhong Li, Yves Grandvalet, and Franck Davoine. Explicit inductive bias for transfer learning with convolutional networks. In *ICML*, 2018. 2, 6

[32] Xingjian Li, Haoyi Xiong, Hanchao Wang, Yuxuan Rao, Liping Liu, and Jun Huan. Delta: Deep learning transfer using feature map with attention for convolutional networks. In *ICLR*, 2018. 3, 6

[33] Dongze Lian, Daquan Zhou, Jiashi Feng, and Xinchao Wang. Scaling & shifting your features: A new baseline for efficient model tuning. *NeurIPS*, 2022. 3, 6, 8

[34] Jiashuo Liu, Zheyuan Hu, Peng Cui, Bo Li, and Zheyan Shen. Heterogeneous risk minimization. In *ICML*, 2021. 3

[35] Xu Luo, Longhui Wei, Liangjian Wen, Jinrong Yang, Lingxi Xie, Zenglin Xu, and Qi Tian. Rectifying the shortcut learning of background for few-shot learning. *NeurIPS*, 2021. 3

[36] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. arXiv preprint arXiv:1306.5151. 6

[37] Sangwoo Mo, Hyunwoo Kang, Kihyuk Sohn, Chun-Liang Li, and Jinwoo Shin. Object-aware contrastive learning for debiased scene representation. *NeurIPS*, 2021. 3

[38] Mohammed Bany Muhammad and Mohammed Yeasin. Eigen-cam: Class activation map using principal components. In *IJCNN*, pages 1–7. IEEE, 2020. 1

[39] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 4

[40] Haolin Pan, Yong Guo, Qinyi Deng, Haomin Yang, Jian Chen, and Yiqun Chen. Improving fine-tuning of self-supervised models with contrastive initialization. *Neural Networks*, pages 198–207, 2023. 3, 6, 8

[41] Judea Pearl et al. Models, reasoning and inference. *Cambridge, UK: CambridgeUniversityPress*, 19:2, 2000. 5

[42] Judea Pearl and Dana Mackenzie. *The book of why: the new science of cause and effect*. Basic books, 2018. 5

[43] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *ICCV*, pages 1406–1415, 2019. 6, 9

[44] Yushan Feng Saihui Hou and Zilei Wang. Vegfru: A domain-specific dataset for fine-grained visual categorization. In *ICCV*, 2017. 6

[45] Krishna Kumar Singh, Dhruv Mahajan, Kristen Grauman, Yong Jae Lee, Matt Feiszli, and Deepti Ghadiyaram. Don't judge an object by its context: Learning to overcome contextual bias. In *CVPR*, pages 11070–11078, 2020. 3

[46] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, pages 776–794, 2020. 4

[47] Lifu Tu, Garima Lalwani, Spandana Gella, and He He. An empirical study on robustness to spurious correlations using pre-trained language models. *Transactions of the Association for Computational Linguistics*, 8:621–633, 2020. 3

[48] Vladimir Vapnik. Principles of risk minimization for learning theory. In *NeurIPS*, pages 831–838, 1992. 3

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017. 5

[50] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 1, 6

[51] Zirui Wang, Zihang Dai, Barnabás Póczos, and Jaime Carbonell. Characterizing and avoiding negative transfer. In *CVPR*, pages 11293–11302, 2019. 1

[52] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018. 6

[53] Kai Yuanqing Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. In *ICLR*, 2020. 3

[54] Tete Xiao, Colorado J Reed, Xiaolong Wang, Kurt Keutzer, and Trevor Darrell. Region similarity representation learning. In *ICCV*, pages 10539–10548, 2021. 3

[55] Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. In *ICCV*, pages 8392–8401, 2021. 3

[56] Jiahao Xie, Xiaohang Zhan, Ziwei Liu, Yew Ong, and Chen Change Loy. Unsupervised object-level representation learning from scene images. *NeurIPS*, 34, 2021. 3

[57] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057, 2015. 5

[58] Kaichao You, Zhi Kou, Mingsheng Long, and Jianmin Wang. Co-tuning for transfer learning. *NeurIPS*, 33, 2020. 3, 6, 7

[59] Yifan Zhang, Bryan Hooi, Dapeng Hu, Jian Liang, and Jiashi Feng. Unleashing the power of contrastive self-supervised visual models via contrast-regularized fine-tuning. *arXiv preprint arXiv:2102.06605*, 2021. 3, 6

[60] Heliang Zheng, Jianlong Fu, Zheng-Jun Zha, and Jiebo Luo. Looking for the devil in the details: Learning trilinear attention sampling network for fine-grained image recognition. In *CVPR*, 2019. 6

[61] Jincheng Zhong, Ximei Wang, Zhi Kou, Jianmin Wang, and Mingsheng Long. Bi-tuning of pre-trained representations. *arXiv preprint arXiv:2011.06182*, 2020. 1, 3, 6, 8

[62] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 6

[63] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 127:302–321, 2019. 6, 9